VOLUMEN I NO.2 FEBRERO 1986

ENLACE

INFORMACION SOBRE EL CLUB

La integración en la asociación C.E.I.U.Q.L. se hace por suscripción anual. El C.E.I.U.Q.L. publica mensualmente el boletín de los socios QLave, nombre por el cual también se conoce al club.

Más información sobre la asociación se puede obtener desde la secretaría del club.

Para ser miembro de QLave se requiere estar interesado en el micro-ordenador Sinclair QL.

El club mantiene una librería de software. Una lista de los programas existentes en la librería se publicará de vez en cuando para así ir actualizándola. Los programas que se quieran aportar o sacar de la librería se deben notificar directamente al encargado de la misma.

Por favor enviad todas las contribuciones a QLave al presidente.

Presidente	Secretario	Librero
Serafín Olcoz	Juan Palacio	Angel Asín
Baltasar Gracián 21,1-cto.	Pedro Mª. Ric 19, 4	San Jorge 22,3-B
50005 ZARAGOZA	50008 ZARAGOZA	50001 ZARAGOZA

Contribuciones a QLave:

Las contribuciones a QLave deben ser Archivos de QUILL en cartuchos de microdrive, preferiblemente. Los cartuchos se devuelven a vuelta de correo.

Los programas cuya extensión no sea muy grande se incluirán en la revista, pero aquellos de gran extensión pasarán a formar parte de la librería.

Los programas que enviéis deben adjuntar una descripción de los mismos y de su funcionamiento.

==========

Se prohibe la reproducción total o parcial del contenido de esta revista.

QLAVE no se hace responsable del contenido de los artículos o comentarios que aparezcan firmados por sus correspondientes autores.

Sinclair, QL, QDOS, ZX microdrive son marca registrada de Sinclair Research Ltd.
QUILL, ABACUS, ARCHIVE, EASEL son marca registrada de PSION Ltd.

EDITORIAL

Aunque algo tìmidamente hemos inaugurado la secciòn 'Correo de los Socios'. A ver si os decidis a scarle el jugo a @Lave enviando vuestras cartas y colaboraciones.

Por fin respondiò el Gobierno Civil. Para que todos sepais cômo està el 'asunto' legal, he dedicado una pàgina exclusivamente para ello.

El pasado 30 recibì una carta de Tony Tebby, diseñador del QDOS, en la que agradecia el hecho de nombrarle socio de honor (creo que desarrollar un sistema operativo para el Q.L. es suficiente mèrito para ello) y me enviaba noticias sobre lo ùltimo en software realizado por @Jump Ltd. En la misma carta me comentaba que estaba en negociaciones con INVESTRONICA S.A., para vender su software en España y ojo a la noticia: en Español.

Por motivos de tiempo al cerrar @Lave no he podido confirmar con INVESTRONICA el estado en que se encuentran dichas negociaciones pero en el pròximo nùmero os hablarè de ello.

Parece ser que se quiere dar salida al Q.L. como ordenador de juegos, el botòn de muestra puede ser el que la Caja de Ahorros de Zaragoza, Aragòn y Rioja lo promocione como tal en su propaganda.

De hecho cada vez hay màs software de aventuras para el Q.L. aunque sea realmente escaso, todavia, en España.

Para aquellos que siguen la seccièn de introduccièn al còdigo màquina decirles, que como pueden ver en èste nùmero llegamos a la introduccièn a la arquitectura del 68008 como se habia anunciado ya en el nùmero 0.

Siguiendo con este tema espero no sòlo vuestra colaboraciòn sino que ademàs espero vuestras criticas al contenido de @Lave, su nivel, etc...

Respecto a las colaboraciones que me envieis para publicarlas en @Lave, debo deciros que lo hagais una semana antes de que finalice el mes, màs o menos, para poder incluirlas en el nùmero siguiente ya que de no ser asi se crean problemas a la hora de hacer el montaje, que aunque parezca que no, lleva lo suyo.

Por supuesto que tambièn espero que mandèis programas para incorporarlos a la libreria. Para aclarar el modo de hacerlo asi como la forma en que se pueden obtener dichos programas, el encargado de la libreria dedica a ello una pàgina en este nùmero.

Para el pròximo nùmero espero tener la oportunidad de daros el nùmero de Asociaciòn con que se haya inscrito @Lave. Una vez que lo tengamos se facilitarà el trabajo del secretario y del Tesorero ya que las transferencias bancarias se podràn hacer directamente a nombre del Club.

Serafin Olcoz

CORREO DE LOS SOCIOS

Un tirón de orejas al autor de la información sobre los 68000, los 68000 fueron presentados honradamente como procesadores de 16 bits, con estructura interna de 32, antes de que algunas muy conocidas firmas de microprocesadores presentasen 16 bits con bus de datos de 8 bits.

Aunque algunos "espabilaos" estan diciendo que son de 16 bits por dentro, eso es mentira, si bien tienen Alus de 16 bits, tienen tres, es decir son de 48 bits. Esto lo hacen porque quieren relegar a los 68000 a la misma categoria que los eqipos de Anticuada Tecnologia de 16 bits, que pretenden ellos son los "buenos".

La diferencia es abisal, aunque el fenomenal Superbasic, (una de las primeras implementaciones aunque incompleta y con pequeñas diferencias del basic STANDARD, con mayúsculas, más información "Standares de la Informàtica" ZX, Diciembre 1985), tiene un problema de velocidad en programas cortos, pero con el Simon Goodwin's Supercharge Superbasic Compiler, se superresuelve.

La razón de la increible diferencia está en el direccionamiento cómodo por encima de 64K, y en su velocidad independiente de la longitud de los programas.

Bueno no hagàis mucho caso a las revistas de informàtica porque tienen muy poquita idea todavia. Aunque algunas ya están aprendiendo.

Jose-M Guzman
Ramon y Cajal 51
41005 SEVILLA

==========

Es una forma de hablar el decir que el MC 68008 es un microprocesador de 16 bits. Realmente como mejor se puede describir, a nuestra manera de ver y , de una forma reducida (èsto es sin entrar en detalles) es como un microprocesador de 32/8 bits.

RESPUESTA DEL GOBIERNO CIVIL

Justo tres dias antes de expirar el mes, plazo en que esperaba la respuesta del Gobierno Civil a la presentaciòn de la Asociaciòn: Club Español Independiente de Usuarios del O.L., èsta se produjo.

En el escrito que ha remitido el Gobierno Civil se me notifica que han recibido mi escrito junto con la documentaciòn relativa al C.E.I.U.O.L. en el que solicitaba suscripciòn en los correspondientes Registros Pùblicos.

Como era de esperar al examinar la documentaciòn aportada, ha observado determinadas deficiencias que ràpidamente se han subsanado siguiendo las indicaciones que para ello se sugerian.

Las deficiencias encontradas son un total de nueve, y aunque todavia no tenèis una copia de los estatutos (no tiene sentido hacerla hasta que no estèn definitivamente aprobados), voy a especificaros en que consistian :

1.- Completar el articulo 6º,haciendo constar entre los requisitos generales para ser socio la mayoria de edad (18 años), independientemente de que si puedan pertenecer a la Entidad personas menores de edad pero con la salvedad de que, tales socios no gozaràn de los derechos de voto ni de ser electores ni elegibles para cargos directivos. Estas limitaciones afectan igualmente a los socios de honor.

2.- Hacer constar en pesetas el limite de presupuesto anual que se prevee, en el articulo 13º.

3.- Completar en el articulo 19º las competencias de la Asamblea General Extraordinaria con lo siguiente: Disposiciòn y enajenaciòn de bienes, constituciòn de Federaciones o integraciòn en ellas, ratificar la expulsiòn de un socio a propuesta de la Junta Directiva y solicitud de utilidad pùblica.

4.- Ampliar, en el articulo 20º, a quince dias el plazo que debe mediar entre la convocatoria de la Asamblea General, tanto ordinaria como extraordinaria, y la celebraciòn de la misma.

5.- Realizar las rectificaciones necesarias en el articulo 22º, para que el quorum de votaciòn de la Asamblea General Extraordinaria, afecte a las competencias que se han ampliado en el 3er punto.

6.- Completar el articulo 26º con la duraciòn del mandato de la Junta Directiva.

7.- Hacer constar en el articulo 28º el quorum de asistencia de la Junta Directiva.

8.- Redactar correctamente el pàrrafo 2º del articulo 35º ya que su redacciòn es confusa.

9.- Al final de los Estatutos, deberàn firmar todos los promotores y no sòlo el presidente y el secretario.

Una vez subsanados estos nueve puntos, el pasado dia 28 remitimos la documentaciòn al Gobierno Civil a fin de proceder a la inscripciòn solicitada.

Espero que, nuevamente, antes de un mes me conteste el Gobierno Civil indicàndome si debemos realizar alguna nueva rectificaciòn (espero que no) o por el contrario si ya se ha realizado la inscripciòn del C.E.I.U.O.L. como asociaciòn.

S.O.Y.

COMENTARIO DE LIBROS

Libro: Guia del Usuario Avanzado del Q.L.
Autor: Adrian Dickens
Editorial: RAMA (Versión Española)
Precio: 2800 pts.

Este libro contiene una descripción completa del QDOS, (uno de los cinco sistemas operativos que por ahora tiene el Q.L.), Orientada fundamentalmente al usuario en código máquina o en lenguajes como el C y el BCPL.

La información es muy completa explicando una por una todas las llamadas al sistema, los TRAPS, para aquellos que procedan del Z-80, son como los restart, pero muchisimo más potentes.

Tiene una muy completa explicación de los utilidades, incluidas en la ROM, y está muy bien organizado de forma que es muy fácil encontar lo que se necesite.

Por cierto uno de los errores más comunes de los programadores es el olvidarse que en las llamadas y traps muchos registros cambian de valor, y eso está siempre indicado. Muchos fallos de programas vienen de esos olvidos.

Además de todas las descripciones de variables sistema operativo, variables intérprete Basic, organización de canales etc., tiene capitulos dedicados al conocimiento de los microdrives, y de la expansión del Basic y del Sistema operativo, expansion de dispositivos, etc.

Es un libro conveniente para todo el que se quiera introducir más allá de los lenguajes habituales, e imprescindible para los programadores en Assembler, C, BCPL. Sin embargo la multitarea, puede ser utilizada con el FORTH 83, u otros lenguajes.

Las principales criticas son : falta una explicación de como utilizar el COMMON HEAP y el USER HEAP, zonas de almacenamiento adicional de datos, y pueden resultar de uso dificil para la mayoria de los usuarios.

Por otro lado los programas en multitarea son un ejemplo de lentitud, debido a la forma de suspenderlos, están la mayoria del tiempo parados, el reloj es muy útil, debido a que apenas le come tiempo al Basic y a otros programas, pero se vuelve lentisimo por su bajisima prioridad cuando hay otro programa ejecutándose.

En el programa de gráficos en multitarea, hay que dividir todos los tiempos por diez, para que funcione bien.

Resumiendo pese a estas criticas menores, Un libro imprescindible para todo el que

quiera ir mas allà. Espero que la versiòn traducida incorpore las nuevas utilidades y traps, I VER$ JS o WGE). Por cierto un libro de ensamblador, se hace necesario para utilizarlo, aunque se puede entender bastante bien sin él.

Jose M. Guzmàn

Nota: La versiòn española del "QL programaciòn avanzada" efectivamente incorpora las utilidades y Traps correspondientes a la versiòn WGE. Lamentablemente ni en la versiòn española de este libro se han incluido las caracterìsticas de la versiòn 1.13 del QDOS, versiòn èsta del sistema operativo que rige la versiòn española del QL.

COMENTARIO DE PROGRAMAS

Programa: GRAPHIQL
Editores: TALENT COMPUTER SYSTEMS
Precio: 10.400 pts.

Este es un programa que nos permite el diseño de dibujos en la pantalla del QL. Trabaja en baja resoluciòn, 256x256 pixels, y ha sido creado por los chicos de TALENT (Doctor, Animate, Backdrop...).

El programa està protegido como tantos otros en el mercado pidiendo el "master" para poder ponerse en funcionamiento, si trabajamos con una copia como se debe hacer por razones de seguridad.

Junto al prograza se nos entrega otra cinta con tres pantallas diseñadas con este paquete y hay que decir que son realmente estupendas si bien el tiempo necesario para hacer nosotros algo semejante es elevado.

Posee la mayor parte de las caracterìsticas comunes a este tipo de programas:

-Podemos ampliar la imagen para obtener una visiòn detallada punto a punto y reducirla posteriormente; puedes tener informaciòn constante de las coordenadas en que estès; comandos que por desplazamiento del cursor dejas una linea de puntos...

Pero veamos algunas caracterìsticas diferenciadoras de este programa:

Podemos dibujar lìneas rectas desplazàndonos con el cursor y al llegar al lugar destinado fijarla, es decir la lìnea nos sigue donde vayamos. Se pueden hacer circulos y elipses cuyo tamaño podemos ver mientras lo incrementamos o decrementamos con un toque de cursor.

Tiene tres tipos de cursores para escoger el que consideremos màs apropiado en cada ocasiòn.

Admite trabajar en dos modos bàsicos: JAM que sustituye el color que hubiese anteriormente en el "papel" por el actual y XOR en el que el nuevo color depende del que tuviese el punto por el que pasemos.

Una de las caracteristicas màs curiosas del programa es el Airbrush, que provoca una serie de puntos aleatorios alrededor del cursor. Es una idea desde luego interesante pero que conviene usar con cuidado y en el lugar oportuno ya que es dificil corregir lo ya realizado, tendriamos que ir cambiando cada uno de los pixels de forma individual.

Posee tambièn la posibilidad de memorizar bloques del dibujo que posteriormente podremos colocar donde lo consideremos oportuno, movièndolo o copiàndolo en otra zona de la pantalla.

Lògicamente podemos hacer reflexiones horizontales, verticales y rotaciones.

Lo màs sobresaliente del programa son las texturas, cuyo funcionamiento es bastante interesante. Si pulsamos el comando TO nos aparecerà superpuesto en la mitad inferior de la pantalla una zona que podemos usar a modo de paleta de pintor en la que podemos realizar una mezcla de puntos a distintos colores, esta zona diseñada asi podrà ser memorizada asignàndole una letra del alfabeto para poder usarla posteriormente en nuestro dibujo. El uso de esto tiene dos modalidades pintar directamente la textura que queramos sobre la pantalla situando el cursor en el lugar oportuno o bien rellenando una determinada zona con ella, èste ùltimo serà el modo màs habitual de trabajo. Decir tambièn que en el programa se incluyen una serie de texturas ya definidas como combinaciòn de colores dos a dos, con lo que se obtienen 64 texturas distintas.

Ya he hablado de la posibilidad de rellenar una una zona con una textura, también se puede rellenar esta zona con un determinado color. Es de agradecer que antes de colorear se haga uso del comando FA con el que se comprueba que no nos hemos ningùn punto por el que se nos pueda escapar el coloreado y sòlo despuès de esta comprobaciòn es cuando se puede hacer uso de los comandos de 'fill'.

Hay otro grupo de comandos destinados a facilitar el trabajo con colores; asi podemos intercambiar el color existente en una determinada zona por otro sin màs que teclearlo directamente: CS01 por ejemplo, intercambia el negro por el azul;podemos alterar el color del cursor; es posible realizar un 'merge' de dos colores aunque hay que hacerlo con cuidado ya que una vez realizado dificilmente podriamos deshacerlo. Una posibilidad màs a tener en cuenta en este contexto del color es la posibilidad de asociar una lista de colores a usar a una textura o bloque, asi podemos incluir o retirar colores de esa lista con un simple comando esta facilidad nos ayudarà en nuestro trabajo con las texturas.

Una posibilidad màs y bastante usual en estos programas es el incluir texto en el dibujo, algo que hay que usar con cuidado por las difucultades de rectificaciòn posterior.

El programa incluye un menù que nos ayuda a gestionar la lectura y grabaciòn de las pantallas realizadas y/o las texturas que hayamos creado, asi mismo podemos formatear un microdrive y obtener su directorio.

Existe ademàs la posibilidad de solicitar las ayudas del programa, con lo que recordaremos los comandos que precisemos. Decir sin embargo que son tal vez algo pobres en informaciòn sobre las funciones de los diversos comandos.

Con el programa viene ademàs una rutina que nos permitirà obtener copias de la pantalla por impresora se trata de una rutina bien hecha y orientada a la posibilidad de alterarla dependiendo del tipo de impresora que poseamos. El problema fundamental que le encontramos es el de su increible lentitud al estar en BASIC su mayor parte, ver que tarda

alrededor de 25 minutos en acabar de sacarla. Asi pues la alternativa es pasarnos la rutina a código máquina, compilándola o reescribiéndola. Respecto al funcionamiento de esta rutina decir que consta de un bloque principal en Superbasic y una pequeña rutina en máquina que se encarga de transformar cada una de las lineas de puntos (256 en total) de la pantalla a la impresora, ésto se hace de tal forma que puede incorporarse una impresora a color ya que el tratamiento dado a la información extraida de la pantalla se refiere a una asignación de color a los bytes de la pantalla, con lo que incluso con una impresora de "blanco y negro" se consigue una interesante gama de grises; ahora bien es preciso haber experimentado un poco con los efectos causados por la impresión ya que por el uso de esta rutina de conversión de colores puede ser que recibamos alguna sorpresa en el proceso de impresión.

Respecto al folleto de instrucciones incluido con el programa decir que es bastante completo. Posee cuatro secciones fundamentales: la primera se refiere a los comandos y posibilidades de éstos. La segunda se ocupa de la inclusión de pantallas en programas BASIC, sección ésta bastante prescindible ya que no aporta nada que el usuario no sepa ya.

En la tercera parte se nos cuenta la rutina de impresión de la pantalla dando las indicaciones precisas para la alteración de ésta. En la última parte se nos hacen una serie de indicaciones acerca de cómo podemos trabajar mejor con el programa desde el punto de vista artistico que no ya desde los comandos ya comentados en otras secciones, además se incluye una sección en la que se nos habla de la fotografía de la pantalla y su aplicación práctica a nuestro caso.

El folleto finaliza con un glosario de las instrucciones y una nota acerca de la posibilidad de la incorporación del programa en un floppy.

En conjunto pues una presentación bastante cuidada respecto al folleto de presentación y por los ejemplos suministrados con el programa.

Respecto al programa en si ya hemos hablado detenidamente de sus comandos individuales, en conjunto adolece de la poca resolución conseguida al trabajar en 256x256 pixels ya comentado anteriormente, aunque este hecho se vea compensado por la vistosidad aportada por el color.

Otra deficiencia observada es la lentitud de ejecución de gran parte de los comandos como puede ser el dibujo de lineas, circulos y elipses que parece realizarse prácticamente a saltos o el desplazamiento del cursor en el proceso de pintura que si bien posee la opción de movimiento rápido no es en modo alguno lo apetecido en un programa de estas caracteristicas.

Y acerca de la rutina de impresión ya me he referido a su lentitud y que si bien es preciso tener en cuenta la compatibilidad de las impresoras igualmente se podria haber hecho un programa en código máquina con la posibilidad de alterar los parámetros diferenciadores de las impresoras; 25 minutos de impresión es un periodo de tiempo dificilmente justificable de cualquier forma que se plantee.

Isidro Asln.

Programa: ALSISTOCK
Editores: ALSI Comercial S.A.
Precio: 18.000 (mdv), 20.000 (disco 3½")

El ALSISTOCK es un programa creado por ALSI Comercial S.A. que como su propio nombre indica se emplea para realizar el control de almacenes.

Al igual que el ALSIMAIL, programa que comentábamos en el número anterior, el ALSISTOCK presenta la pantalla dividida en dos secciones:

- Zona de mensajes

- Pantalla principal

Como presentan el mismo formato que las del ALSIMAIL, no merece la pena volver a especificarlas.

Al comenzar a trabajar es preciso tener además del ALSISTOCK en el mdv1, un cartucho formateado en el mdv2. Ambos cartuchos deben permanecer todo el tiempo de trabajo en sus lugares correspondientes debido a que el programa accede a ellos en diversos momentos durante el período de trabajo.

Nada más cargarse el programa aparece en pantalla el menú que permite abrir ficheros (hasta 10 ficheros diferentes), borrar ficheros, cargar datos, actualizar y conocer el margen general de beneficios.

Una vez abierto un fichero, como mínimo, al seleccionar la opción de cargar datos hay que especificar el número del fichero que se va a utilizar para seguidamente acceder al menú principal que consta de las opciones: Modificación, introducción, búsqueda, borrado,cambio de ficheros, actualización.

Mediante la opción de modificación de artículos se puede variar cualquiera de los artículos que residen en memoria, (código, nombre, cantidad o precio). Así mismo se puede modificar el precio de todos los artículos (o de un código a otro) variando el porcentaje esto es aumentándolo o disminuyéndolo.

Para evitar posibles errores, al realizar una modificación el programa pregunta si es correcto o no lo es, para, en caso de que sea incorrecto, modificar de nuevo los datos.

La opción de listados da paso al menú: artículos, dir mdv1, dir mdv2, vuelta al menú principal.

Por supuesto y como es característico en los programas de control de almacenes, al seleccionar la opción artículos se podrá elegir entre todos o simplemente aquellos que se encuentren bajo mínimos.

En cualquiera de ambos casos se pasa al submenú que ofrece las posibilidades de seleccionar todos, de código a código o la muy acertada opción de búsqueda fraccionada.

La opción de búsqueda fraccionada es una característica muy útil de este programa ya que permite seleccionar todos aquellos artículos que tienen en común una fracción de su

código. Como cada código tiene nueve dígitos en ellos se puede incluir además de la fecha otras características del artículo. De este modo se da un amplio márgen a la forma en que se pueden obtener los listados de los artículos almacenados.

La opción de listados finaliza preguntando si estos se desean con o sin precio de costo. Posteriormente el listado se puede sacar por pantalla o impresora especificándose en él, el número de orden, código, concepto, precio, cantidad alsina en almacén, precio total y precio de coste. Aunque también se pueden limitar los listados de modo que sólo se especifique el precio y la cantidad para así poder emplearlos como listas de precios; o bien seleccionar sólo los artículos que están bajo alsina o desde un código a otro.

La opción de introducción informa de los artículos que residen en memoria así como también permite introducir artículos nuevos.

Por supuesto que el cambio de fichero lo que hace es permitir el acceso a otro fichero diferente durante la utilización del programa sin alterar para nada el funcionamiento del mismo. Aunque la versión que probamos (versión I) carecía de esta opción ya que se trata de una modificación posterior, su funcionamiento es el descrito.

Por último queda la opción de actualización. Al acabar de utilizar el programa, se debe seleccionar la opción 0 que graba los nuevos artículos introducidos siempre y cuando se hayan producido altas o bajas de artículos o ficheros.

También realiza la suma de los totales en almacén.

Como se puede ver por la descripción realizada anteriormente de las características este programa cumple con todas las condiciones requeridas para realizar el control de almacenes.

Destacando la ventaja de poder realizar los listados por medio de una búsqueda fraccionada del código de los artículos. Así como la posibilidad de realizar una revalorización sobre el precio del artículo (en %).

Los problemas que plantea son: el no poder asociar varios precios a un mismo artículo, el que no calcule el Precio Medio Ponderado (con sus tres modalidades), coste estándar..., así como que no permita el control del almacenaje de un artículo en diferentes almacenes.

Para una perfecta gestión se debe emplear junto con el ALSISTOCK, de forma que se intercomuniquen, un programa de facturación así como otro que proporcione el código y todos los datos referentes a los proveedores.

Serafín Olcoz

LOS 68000:

UNA FAMILIA DE MICROPROCESADORES

Los microprocesadores de la familia 68000, se iniciaron por el año 1979, en que se presentó el primero, el 69000, llamado así por contener alrededor de 68000 transistores, siendo el primer microprocesador presentado al mercado, con una arquitectura de 32 bits y una estructura interna de 32 bits.

Motorola, el fabricante, desarrolo un procesador totalmente nuevo, incompatible con sus anteriores, realizando una arquitectura imitada de los IBM 370, y siguiendo una evolución similar a la de éstos.

En estos momentos existen ya ¡ cinco ! versiones de los procesadores de la familia, y como todos sabemos que el precio de los chips desciende continuamente, por lo tanto los procesadores que ahora son inasequibles, los podremos tener en el futuro a costes razonables.

Por ello vamos a empezar por el más potente de todos, el 68020, aunque calificado como un auténtico 32 bits, es más que eso, realmente sólo existe otro procesador con la misma potencia, el ATT 32100. Para los más entendidos, los dos tienen una pipeline de tres niveles, con 32 bits en cada nivel, esto permite normalmente estar escribiendo el resultado de una operación, al mismo tiempo que se calcula la siguiente y se toma el dato o la instrucción posterior, aunque la longitud de la pipeline de estos dos procesadores es inferior a la de otros de 16 bits, esto es debido a que la pipeline tiene un tamaño óptimo, cuando se producen saltos o interrupciones, la pipeline se vacía y hay que volver a rellenarla, con lo que si es muy grande se pierde al final más tiempo del que se gana.

Los dos tienen la pipeline muy apoyada en hardware, con ¡ varias ! 'alus' de 32 bits. El 68020 tiene una ALU para las operaciones, otra para calculo de direcciones de datos y otra para cálculo de direcciones de instrucciones, el hecho de tener varias 'alus' les permite reconfigurarlas y trabajar sobre 64 bits.

Los dos tienen una 'caché' memoria interna, de instrucciones, donde se almacenan 256 bytes de instrucciones, 64 palabras de 32 bits, esto les da una gran potencia en operaciones repetitivas, dado que acceden a la vez a la instrucción en la caché y al dato en memoria.

Obviamente tienen coprocesadores numéricos, pero implementados al estilo de los grandes ordenadores y superminiordenadores, como extensiones de la arquitectura y no como periféricos, espacios de direccionamiento de 32 bits, osea 4 Gigabytes, memoria y máquina virtual etc. Un detalle fundamental para conocer la diferencia entre estos y los de 8 y 16

bits, es que el espacio de direccionamiento es lineal, sin segmentos ni otros incordios como conmutación de bancos, etc, y que este espacio es por tarea, o grupo de tareas según el sistema operativo, esto puede cambiar incluso a las técnicas de programación pero eso sería un tema para otro capítulo. Manipulan los 4 gigabytes con la misma facilidad que nuestro querido Spectrum sus 64 K.

El inconveniente actual es el coste, un 68020 cuesta lo que un 8L ¡antes de la bajada de precio!, pero esto es porque es muy nuevo, pero ya bajará.

También son muy rápidos; el 68020 tiene tres versiones, con relojes de 12,5 Mhz, 16,6 y la anunciada, pero no presentada todavía de 24 Mhz, realmente los problemas de velocidad de memorias, jerarquía de éstas, implementación de máquinas virtuales, etc. van a ser muy grandes.

Obviamente la puesta a punto va a ser muy larga, pero se puede estar seguro que en el Metalab, como en Apple, Atari, Comodore etc. se estará ya "cociendo" el 68020.

Sin embargo hay que destacar una preciosa característica del 68020, su anchura dinámica del bus, esto le permite trabajar con ROM, RAM y periféricos tanto de 32 bits, como es obvio como de ¡8 y 16 bits!, además los soporta de cualquier velocidad, esto es extraordinariamente importante para facilitar la portabilidad, no tan sólo del software, sino también de los periféricos y accesorios. Hay que comprobarlo, pero resulta muy atractivo, y Motorola hasta ahora siempre ha cumplido.

Siguiendo para abajo en potencia tenemos las dos máquinas virtuales, el 68010 y su versión expandida el 68012, estos ya son de bus de datos de 16 bits, su estructura interna es similar, pero más simple, con pipelines relativamente cortas, y con la ventaja respecto a los más pequeños de su modo bucle, este modo es que cuando detectan un bucle corto, mantienen las instrucciones del bucle dentro del pipeline, accediendo sólo a los datos en memoria, teniendo velocidades impresionantes, la estructura interna es la tradicional de los pequeños, con tres unidades de cálculo de 16 bits, trabajando a la vez y con sistemas de comunicación entre ellas, el hecho de tener tres es para acelerar la pipeline calculando sobre datos y direcciones a la vez, la comunicación entre las unidades de cálculo ha sido mejorada, respecto a los más pequeños, acelerándose los cálculos sobre 32 bits.

La diferencia entre el 68010 y el 68020 es básicamente que el primero tiene un espacio de direccionamiento de 16 Megabytes y el segundo de 2 Gigabytes.

Por último los dos pequeños, el 68000 y el 68008, no soportan máquina virtual, aunque APPLE en el Lisa consiguió una implementación parcial de la memoria virtual, su estructura interna es la misma, con sus tres "alus" de 16 bits, la diferencia con los 68010 es la ausencia de modo bucle, y una mayor lentitud en operaciones de 32 bits.

El 68000 tiene un bus de datos de 16 bits, y un direccionamiento de 16 Megabytes, el

68008 un bus de datos de 8 bits, y un direccionamiento de 1 Megabyte, la diferencia en velocidad de proceso no es tan grande como se dice, no tengo referencias exactas, pero si vale como comparación la direcencia entre un 8088 y un 8085 es de un 30 a un 35%, a la misma velocidad efectiva del reloj, dado que los dos tienen la misma estructura interna, sólo se diferencian en el bus de datos.

La popularidad de estos micros se está incrementando extraordinariamente, en realidad lo más atractivo para los expertos ha sido su similitud con los IBM 370, la arquitectura con mayor éxito en el mundo de los grandes ordenadores, hasta tal punto son similares que IBM tiene una versión especial de los 68000, a la que se ha modificado la ROM interna que determina el juego de instrucciones, teniendo estas versiones el juego de instrucciones 370 y los registros 370. Como es obvio por razones de Copyright, Motorola no ha colocado este juego de instrucciones en sus procesadores de venta libre, es una pena.

Los 68000 tienen una larga historia detrás de ellos, fundamentalmente como miniordenadores y como multiusuario, como superpotentes estaciones de trabajo de ingenieria y como máquinas para Inteligencia artificial, disponiendo de alrededor de la docena de diferentes sistemas operativos. Modelos como IBM 9000, la serie 200 de Hewlet-Packard, o los equipos de inteligencia artificial de Tektronic son ejemplos de ello, también se utiliza masivamente el Unix y otros sistemas multiusuario.

El Apple Macintosh fue el primero en colocarse a niveles de precio de Ordenador Personal, y el 9L el primero en ponerse a tiro de nuestros hispánicos bolsillos, pero va a seguir la marcha con más, mejores y más potentes modelos de todos los fabricantes.

En estos momentos la carrera de los 68000 en el mercado de microordenadores de precio asequible no ha hecho mas que empezar, pero los modelo que ahora conocemos no son mas que los primeros, va a seguir de momento hasta el 68020, pero dado que los registros de usuario son los mismos en todos, que los mecanismos de direccionamiento son los mismos, y que gracias a la máquina virtual, si hace falta, te pueden los grandes colocar una máquina virtual por tarea, emulando para cada tarea una máquina distinta, y pudiendo emular varias máquinas a la vez en multitarea, se tiene garantizada la portabilidad del soft, hasta el 68020, por ahora, claro que en Motorola, seguirán sacando más microprocesadores.

Realmente nadie puede evitar que su hardware envejezca, pero es necesario el poder conservar el software, lo más costoso en estos momentos, y la historia de la informática muestra que siempre se ha podido conservar el software de los 32 bits.

No incluyo referencias, dado que todas están en Inglés, si hay alguien interesado, que me escriba y le daré las referencias.

José M. Guzmán

TONOS Y MUSICA EN EL Q.L.

Las posibilidades de generar sonidos en el Q. L. están limitadas si las comparamos con otros micro-ordenadores, menos capacitados incluso, debido a que sólo tiene un canal frente a algunos otros que tienen tres. Sin embargo y quizá para compensar, la función BEEP está muy bien desarrollada.

El sonido se genera desde SuperBasic por medio la función BEEP activando las funciones de sonido conformadas en el interior del Q.L.. La función BEEP puede aceptar un número variable de parámetros que dan muchos niveles de control sobre el sonido producido, Gracias a los cuales se puede alterar la calidad del sonido del altavoz incorporado que desgraciadamente es muy pobre, que no es sino un zumbido. Las especificaciones mínimas necesitan únicamente indicar una duración y una frecuencia.

Los ocho parámetros que especifican el sonido a generar son: duración, tono, segundo tono, grad_x, grad_y, envolvente, emborronamiento y componente aleatoria.

El tono especifica la frecuencia del sonido,correspondiendo 1 a una frecuencia alta y por el contrario 255 corresponde a una frecuencia baja.

El problema reside en seleccionar los tonos correspondientes a sus octavas entre los 255 tonos posibles.

Para facilitar esta tarea se pueden emplear los valores sugeridos por Ken Huber (publicados en QUANTA en Febrero de 1985):

124 DO (bajo)	57 DO
116 DO# / RE b	54 DO# / RE b
108 RE	50 RE
102 RE# / MI b	47 RE# / MI b
96 MI	44 MI
91 FA	41 FA
86 FA# / SOL b	38 FA# / SOL b
81 SOL	35 SOL
76 SOL# / LA b	32 SOL# / LA b
71 LA	30 LA
66 LA# / SI b	28 LA# / SI b
61 SI	26 SI
57 DO	24 DO (Alto)

Respecto a la duración de los tonos Ken Huber sugiere 16000 para una blanca, 8000 para una negra, 4000 para un corchea etc...

Como ejemplo del empleo de esta selección de tonos, véase el siguiente programa:

```
100 REMark Program To Play ' The Saints'
110 REMark Ken Huber 12/1/85
120 REMark
130 MODE 512
140 WINDOW #0,512,256,0,0
150 CLS #0
160 WINDOW #2,454,184,32,10
170 WINDOW #1,454,184,32,10
180 WINDOW #3,454,50,32,200
190 PAPER #2,0:INK #2,7
200 PAPER #1,2:INK #1,7
210 PAPER #0,0:INK #0,7
220 BORDER #0,1,2
230 CSIZE #2,0,0
240 CSIZE #1,2,0
250 CSIZE #0,0,0
260 CLS #1:CLS #2:CLS #0
270 PRINT #1, 'Hit any Key to Start Tune ...'
280 a$=INKEY$:IF a$=' ' THEN GO TO 280:END IF
290 RESTORE
300 READ a
310 FOR z=1 TO a
320  READ c,d,e
330  BEEP c,d
340  PAUSE e
350 END FOR z
360 DATA 32,4000,56,12,4000,44,12,4000,41,12,16000,35,60
370 DATA 4000,56,12,4000,44,12,4000,41,12,16000,35,60
380 DATA 4000,56,12,4000,44,12,4000,41,12,10000,35,24
390 DATA 10000,44,24,10000,56,24,10000,44,24,16000,50,60
400 DATA 4000,44,10,4000,44,10,4000,50,10,10000,57,40,4000,57,12
410 DATA 8000,44,22,8000,35,22,4000,35,12,16000,41,60
420 DATA 4000,44,10,4000,41,10,8000,35,22,8000,44,22,8000,57,22
430 DATA 8000,50,22,20000,57,60
440 PRINT
450 PRINT 'Good bye'
460 STOP
```

Nótese el empleo de la función PAUSE después de cada nota, para hacerse una idea precisa de su efecto pruébese a ejecutar el programa sin los PAUSE y obsérvese el efecto desastroso que se produce.

Posteriormente, en Marzo del mismo año, Bill Conhig publicó en SUANTA otra selección alternativa basada en la de Ken Huber pero intentando suavizar el paso entre los tonos.

Según Bill Conhig, si el número del tono fuese proporcional a su frecuencia, el paso habría sido de 2:1 y los intervalos serían la raiz doceava de 2. Para obtener su selección empleó el siguiente programa:

```
1000 s=1/12:p=124/57                2000 s=1/12:p=57/24
1010 FOR N=0 TO 12                  2010 FOR N=0 TO 12
1020  PRINT 57*p^(N*s)!" ";        2020  PRINT 24*p^(N*s)!" ";
1030 END FOR N                      2030 END FOR N
```

De este modo las doce notas seleccionadas serían:

124	57
116.2	53.0
108.9	49.4
102.1	45.9
95.7	42.7
89.7	39.7
84.1	37.0
78.8	34.4
73.9	32.0
69.2	29.8
64.9	27.7
60.8	25.8
57	24

Aunque, como Bill Conhig mismo indica, la precisión de la función DEEP no justifica el uso de números decimales ya que el B.L.redondea cada número decimal a un número entero de modo que por ejemplo 55.49 equivale precisamente al mismo tono que 54.51 y lo que es proporcionalmente peor, 23.51 es lo mismo que 24.49.

Espero que estas selecciones de tonos sean de utilidad.

Nota: La dirección del domicilio social de SUANTA es: 24 Oxford Street,Stony Stratford,Milton Keynes MK11 1JU, United Kingdom.

S.O.Y.

LENGUAJE MAQUINA PARA EL QL (II)

4.- LENGUAJE ENSAMBLADOR Y LENGUAJE MAQUINA

Ya hemos dicho que el 68008 sólo es capaz de comprender las instrucciones escritas en su propio lenguaje: el código máquina.

Una instrucción en código máquina se reduce a un número binario de 16 bits (1000111001110101 representa una instrución en código máquina).

El hacer un programa usando directamente los códigos máquina del 68008, resulta francamente tedioso, además de ofrecer muchas posibilidades para cometer errores. Por eso se usan los programas ensambladores, que asignan a cada código de operación un nemónico, lo cual facilita enormemente la tarea, ya que no hay que componer el programa con los números binarios que representan las distintas instrucciones sino con los nemónicos correspondientes.

Así por ejemplo la instrucción 1000111001110101 se representa con el nemónico RTS.

Una vez terminado el listado de nemónicos y operandos que compongan la rutina que queramos formar, el programa ensamblador se encargará de pasarlo a código máquina.

Al listado que confeccionamos con la ayuda de un ensamblador se dice que está hecho en lenguaje ensamblador y constituye el programa fuente.

Una vez que el programa ensamblador ha transformado los nemónicos y los operandos en los respectivos códigos máquina, al programa obtenido se le llama programa objeto, el cual ya puede ser leido por el microprocesador.

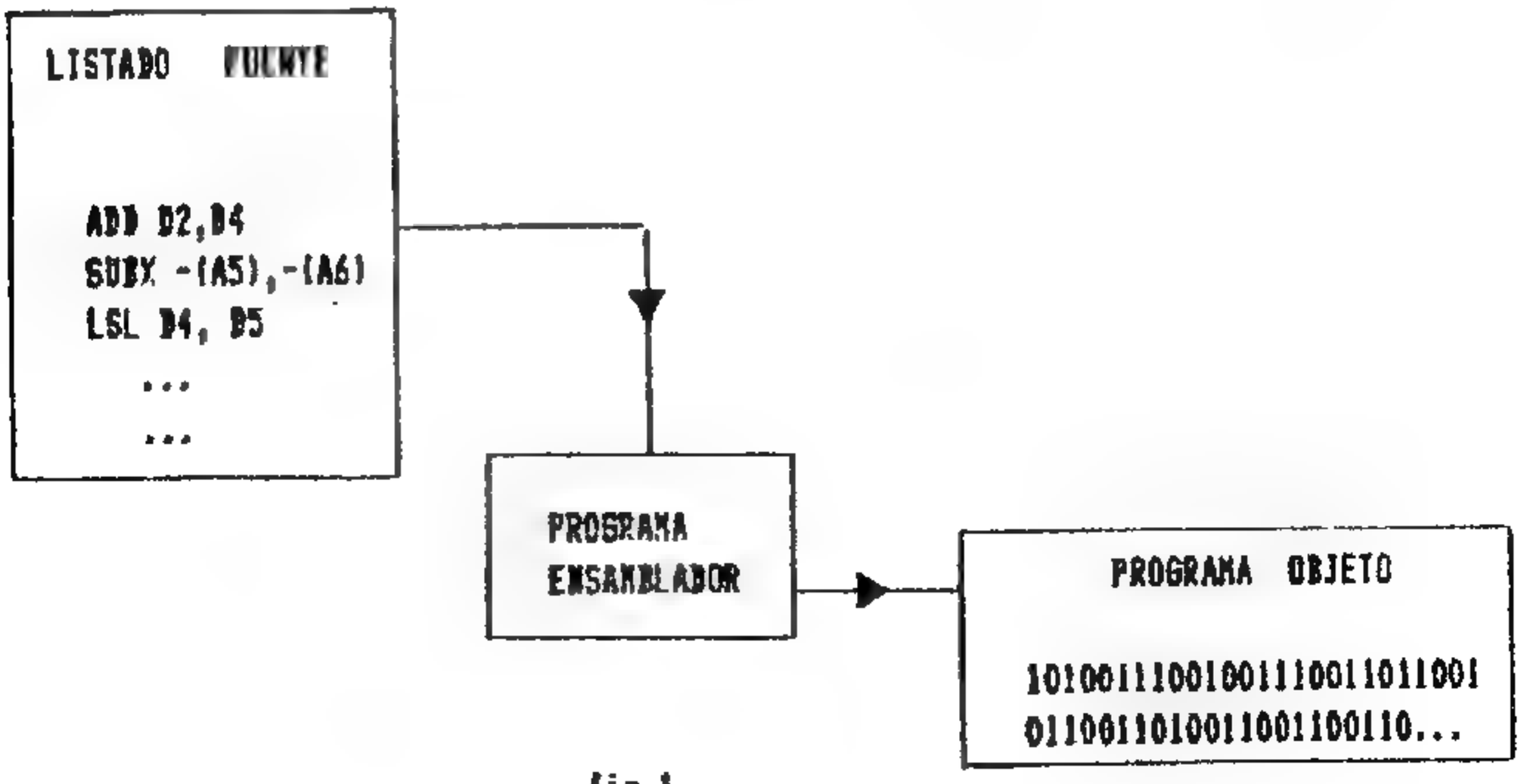


fig.1

A continuación se muestra una lista con todos los nemónicos de las instrucciones el 68000, junto con una breve descripción de las mismas, la notación que se usa en lenguaje ensamblador para representarlas, y los tamaños de operandos que admiten (8, 16 ó 32 bits).

Hemos dividido las instrucciones del 68000 en 3 grupos, separando las instrucciones de uso más frecuente de las que suelen emplearse menos.

1.- Instrucciones usadas frecuentemente

ADD	ADICION	ADD D(n),(ea) / ADD (ea),D(n)	8,16,32
AND	OPERACION LOGICA AND	AND D(n),(ea) / AND (ea),D(n)	8,16,32
ASL	DESPL. ARITM. IZDA.	ASL Dx,Dy / ASL (ea)	
		ASL # (dato), Dy	8,16,32
ASR	DESPL. ARITM. DCHA.	ASR Dx,Dy / ASR (ea)	
		ASR # (dato), Dy	8,16,32
Bcc	BIFURCACION CONDICC.	Bcc (etiqueta)	8,16
BRA	BIFURCACION	BRA (etiqueta)	8,16
BSR	BIF. A SUBRUTINA	BSR (etiqueta)	8,16
CLR	LIMPIA OPERANDO	CLR (ea)	8,16,32
CMP	COMPARA	CMP (ea),Dn	8,16,32
EOR	"OR" EXCLUSIVO	EOR Dn,(ea)	8,16,32
JMP	SALTO	JMP (ea)	--
JSR	SALTO A SUBRUTINA	JSR (ea)	--
LSL	DESPLAZ. LOGICO IZDA.	LSL Dx,Dy / LSL (ea)	
		LSL # (dato),Dy	8,16,32
LSR	DESPLAZ. LOGICO DCHA.	LSR Dx,Dy / LSR (ea)	
		LSR # (dato),Dy	8,16,32
MOVE	DESPLAZA DATOS	MOVE (ea),(ea)	8,16,32
OR	OP. LOGICA "OR"	OR (ea),Dn / OR Dn,(ea)	8,16,32
ROL	ROTACION IZDA.	ROL Dx,Dy / ROL (ea)	
		ROL # (dato),Dy	8,16,32
ROR	ROTACION DCHA.	ROR Dx,Dy / ROR (ea)	
		ROR # (dato),Dy	8,16,32
RTS	VUELVE DE SUBRUTINA	RTS	--
SUB	RESTA	SUB (ea),Dn / SUB Dn,(ea)	8,16,32

2.- Instrucciones poco usadas

ABCD	SUMA EN "BCD"	ABCD Dy,Dx / ABCD -(Ay),-(Ax)	8
BTST	PRUEBA DE UN BIT	BTST Dn,(ea)	
		BTST # (dato),(ea)	8,32
DBcc	CONDICION, DECREMENTO Y BIFURCACION	DBcc Dn,(etiqueta)	16
EXG	INTERCAMBIO REGISTROS	EXG Ry, Rx	32
MOVEM	MUEVE REG. MULTIPLES	MOVEM (lista registro),(ea)	
		MOVEM (ea),(lista registro)	16,32
MOVEP	MUEVE DATOS A PERIFERIC.	MOVEP Dx,d(Ay)	16,32
		MOVEP d(Ay),Dx	16
MULS	MULTIPLICA CON SIGNO	MULS (ea),Dn	16
MULU	MULTIPLICA SIN SIGNO	MULU (ea),Dn	8,16,32
NEG	RESTA DE 0	NEG (ea)	--
NOP	NO OPERA	NOP	
ROXL	ROT. IZDA. CON BIT X	ROXL Dx,Dy / ROXL (ea)	8,16,32
		ROXL #(dato),Dy	
ROXR	ROT. IZDA. CON BIT X	ROXR Dx,Dy / ROXR (ea)	8,16,32
		ROXR #(dato),Dy	
RTE	RETORNA DE EXCEPCION	RTE	--
RTR	RETORNA Y RESTAURA CC.	RTR	--
SBCD	RESTA EN "BCD"	SBCD Dx,Dy	8
		SBCD -(Ay),-(Ax)	--
STOP	PARA Y CARGA DEL SR	STOP # XXX	8,16,32
TST	PRUEBA OPERANDO	TST (ea)	

Aunque existen algunas normas comunes, no todos los programas ensambladores usan la misma "sintaxis". Generalmente las instrucciones en lenguaje ensamblador se dividen en varios campos que suelen ser: Campo de etiquetas, Campo de código o nemónico de operación, campo de operandos o direcciones y campo de comentarios. Los campos de etiquetas y comentarios son opcionales. Así pues unas instrucciones editadas en un programa ensamblador podrían quedar así: (v. Mave n.1 pág 20)

Campo etiqueta	Campo nemónico	Campo oper.	Campo comentarios
TOTAL:	DC.W	$421B	Llevamos total a 421B
RUTINA	MOVE	TOTAL,D2	Lo cargamos en D2

Omitimos el detallar cuales suelen ser los separadores, campos y en general reglas de sintaxis ya que todos los programas las detallan en sus instrucciones, además de que no todos tienen las mismas posibilidades.

3.- Instrucciones rarazente usadas

BCHG	PRUEBA Y CAMBIA UN BIT	BCHG Dn,(ea)	
		BCHG # (dato),(ea)	8,32
BCLR	PRUEBA Y BORRA UN BIT	BCLR Dn,(ea)	
		BCLR # (dato),(ea)	8,32
BSET	PRUEBA Y PONE A 1 UN BIT	BTST Dn,(ea)	
		BTST # (dato),(ea)	8,32
CHK	PRUEBA REG. ENTRE LIMIT.	CHK (ea),Dn	16
DIVS	DIVISION CON SIGNO	DIVS (ea),Dn	16
DIVU	DIVISION SIN SIGNO	DIVU (ea),Dn	16
EXT	EXTENSION DE SIGNO	EXT Dn	16,32
LEA	CARGA DIRECC. EFECTIVA	LEA (ea),An	32
LINK	ENCADENA Y ASIGNA	LINK A # (despiazamiento)	--
NBCD	RESTA DECIMAL CON 0	NBCD (ea)	8
PEA	METER AL STACK DIRECC.	PEA (ea)	32
RESET	RESTEA ELEMENT. EXTERNOS	RESET	--
Scc	PONE A UNO SEGUN CONDIC.	Scc (ea)	8
TAS	PRUEBA Y PUENTE	TAS (ea)	8
TRAP	TIPO "TRAP"	TRAP # (vector)	--
TRAPV	TRAP SI OVERFLOW	TRAPV	--
UNLK	DESENCADENA	UNLK An	--

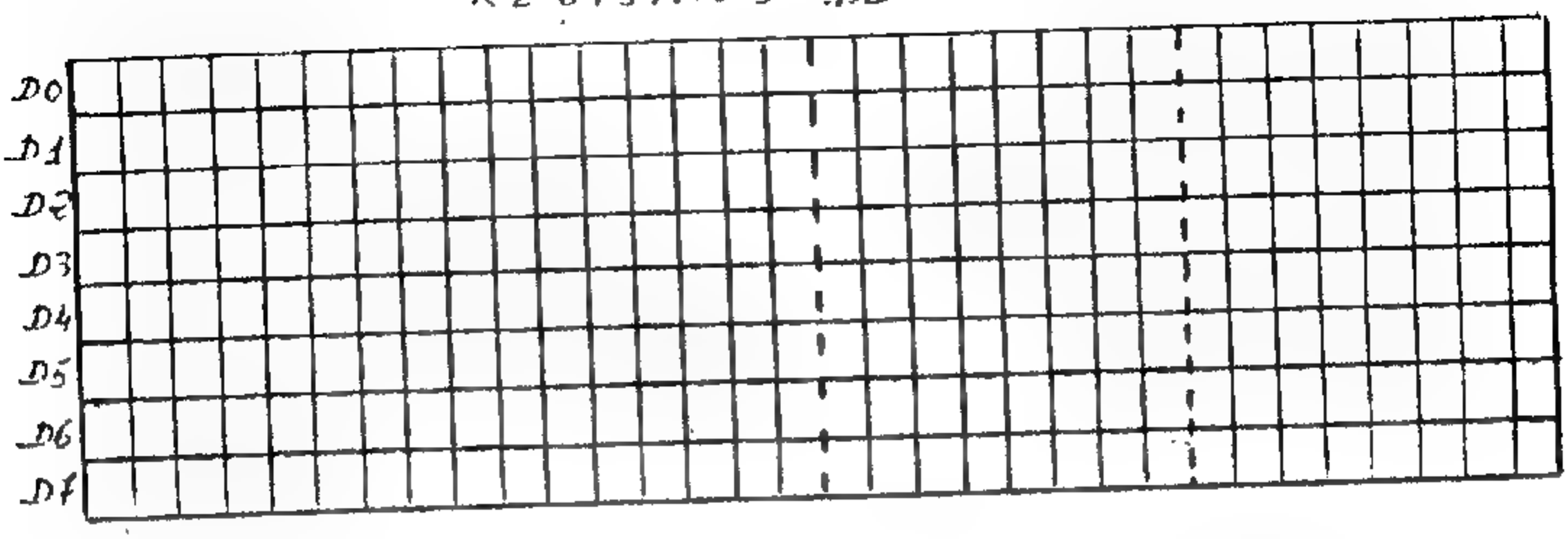
5.-M O D O U S U A R I O Y M O D O S U P E R V I S O R

El 68008 puede trabajar en "modo supervisor" o en "modo usuario". Como veremos más adelante, hay un bit en el registro de estado que determina uno u otro modo.

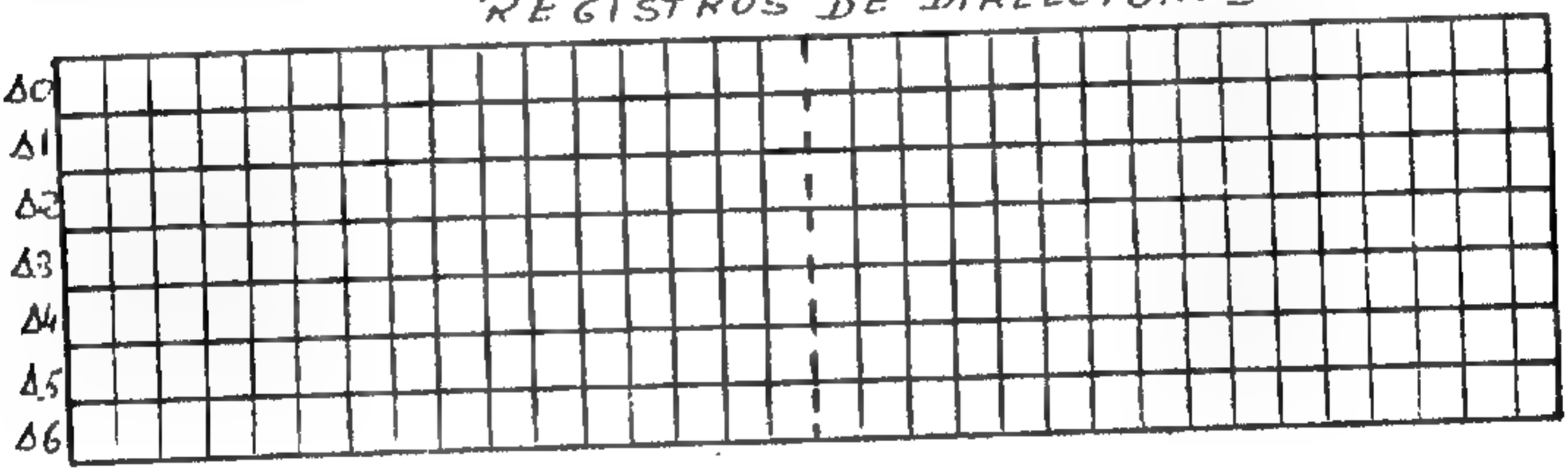
Una de las diferencias entre dichos estados es el que no todas las instrucciones pueden ejecutarse en modo usuario.

También veremos que hay dos punteros de stack, uno para cada modo de trabajo; de forma que si estamos en modo usuario y cargamos un dato en el stack, y acto seguido pasamos a modo supervisor, al sacar el último dato que se ha almacenado en el stack, veremos que no es el mismo que acabamos de meter. Así pues, cuando nos referimos al puntero del stack para trabajar con él, no es necesario que matiCemos si queremos acceder al puntero usuario o al supervisor, ya que en cada momento sólo trabaja con uno, y depende de si el 68008 se encuentra en uno u otro modo, el que trabaje con el puntero usuario o con el supervisor.

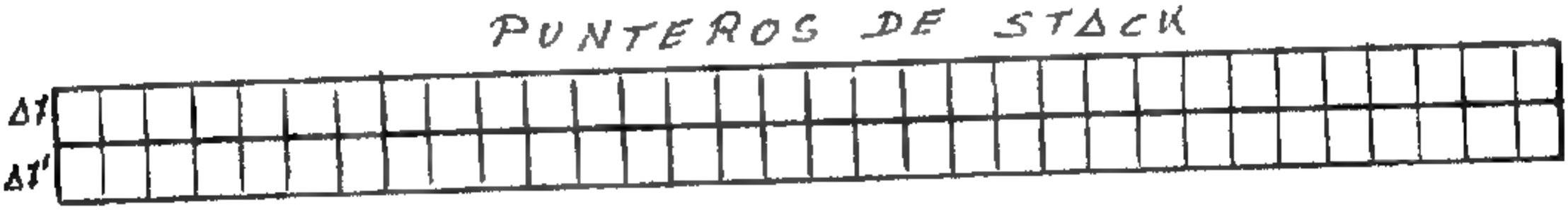
REGISTROS DE DATOS



REGISTROS DE DIRECCIONES



PUNTEROS DE STACK



CONTADOR DE PROGRAMA

REGISTRO DE ESTADO

6.- NOTACIONES USUALES EN LENGUAJE ENSAMBLADOR

Estos son los sìmbolos usados para representar abreviadamente los registros (pàrr. 7), y otras informaciones de uso frecuente en las instrucciones.

An	Registro de direcciones (n indica el nùmero del registro)
Dn	Registro de datos (n indica el nùmero del registro)
Rn	Registro (tanto de datos como de direcciones)
USP	Registro puntero del stack usuario
SSP	Registro puntero del stack supervisor
SP	Registro puntero del stack (tanto usuario como supervisor)
PC	Registro contador de programa
SR	Registro de estado
CCR	Còdigo de condiciòn de los "flags" del SR
EA	Direcciòn efectiva del operando

7.- QUE SON LOS REGISTROS

Todos los microprocesadores tienen lo que podrìamos llamar, unas celdillas de memoria propias, las cuales usan para leer los datos con los que tienen que operar, o para depositar en ellas los resultados de las operaciones. Dichos "lugares" se denominan REGISTROS y no tienen nada que ver con las celdillas propiamente dichas de la memoria del ordenador. Puede claro que no son por lo tanto unas direcciones de memoria, sino registros propios del microprocesador capaces de almacenar datos.

Los registros suelen dividirse en registros de datos, direcciones, de estado... segùn la funciòn de los nùmeros en ellos almacenados.

8.- ARQUITECTURA DEL 68008 ATENDIENDO A LOS REGISTROS DE PROGRAMACION

En la fig. 2 aparecen todos los registros que intervienen en la programaciòn del 68008.

En primer lugar aparecen los 8 registros de datos (D0, D1, D2, D3, D4, D5, D6, D7, D8). Estos registros, como su nombre indica, tienen la finalidad de contener datos u operandos. La informaciòn procesable que puede contener un registro de datos puede estar formada por 8, 16 ò 32 bits, usàndose, segùn los casos los 8 primeros bits, los 16, o el registro completo (fig. 3)

A la hora de programar en lenguaje ensamblador, para determinar el tamaño de la información que se utiliza en un un registro de datos, se coloca detràs del nemònico de la instrucción una B, W ò L (Byte, Word, Long Word), segùn se trate de 8, 16 ò 32 bits, respectivamente.

Por ejemplo, el nemònico de la operación suma es ADD, y por lo tanto la instrucción ADD.B D2,D3 indica la suma de 2 datos de 8 bits contenidos en los registros D2 y D3.

Las operaciones que se llevan a cabo con los registros de datos sòlo afectan a los bits señalados en la instrucción, quedàndose el resto tal y como estaban. Asi, si almacenamos el resultado de una suma en un registro de datos, y dicho resultado tiene una longitud de 16 bits, los otros 16 bits del registro no se alteran.

Esto aunque pueda parecer obvio no siempre ocurre, ya que con los registros de direcciones se afecta todo el registro, aunque el tamaño del operando no sea de 32 bits.

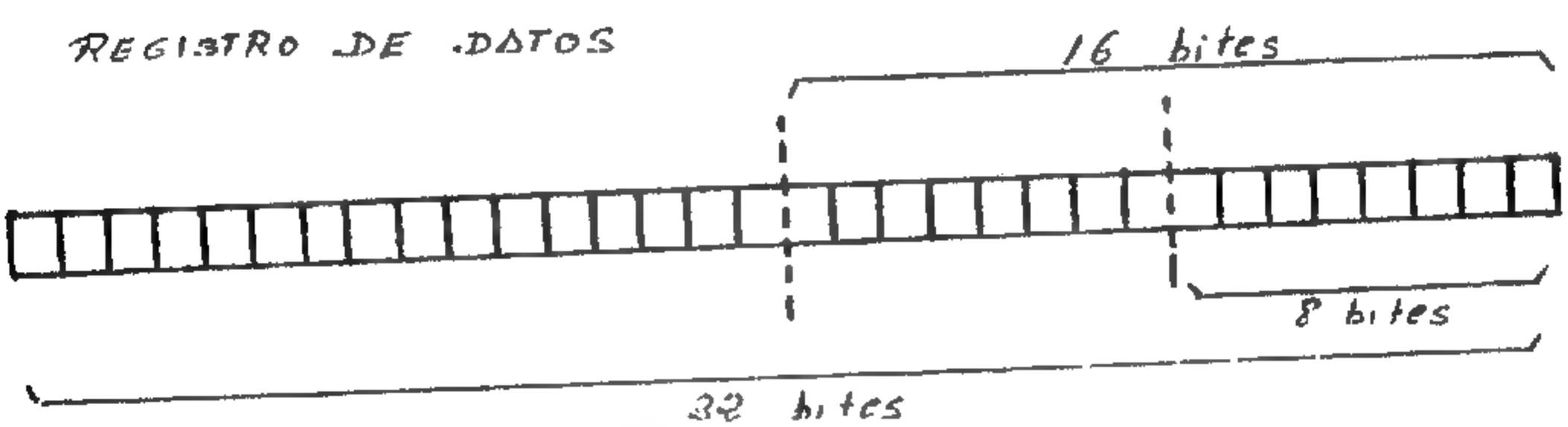


fig.3

Hay 7 registros de direcciones que pueden operar con palabras (16 bits) o con palabras largas (32 bits). Los registros de direcciones no pueden trabajar con bytes.

Como su nombre indica, estos registros se usan normalmente para indicar direcciones de memoria.

Hay que tener en cuenta, que tal y como se muestra en la fig. 4 la memoria del QL está formada por celdillas de 8 bits, con lo cual para almacenar una palabra se usan 2 celdillas y para una palabra larga 4, por lo que las direcciones de memoria correspondientes a palabras han de ser pares, y las referentes a palabras largas se suceden de 4 en 4.

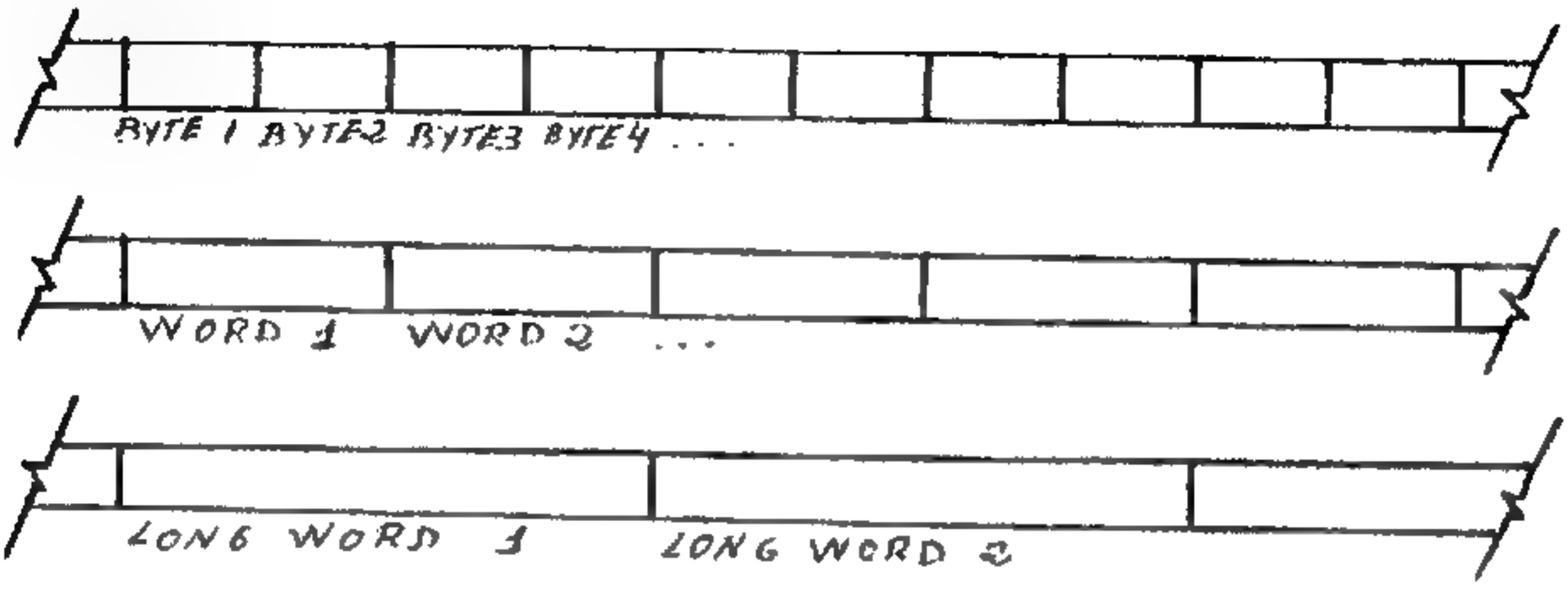


fig. 4

Cuando un registro de direcciones funciona como operando destino, se afectan todos los bits del mismo, aunque el tamaño del resultado sea de 16 bits.

Los 16 bits de <u>mas peso</u> repiten el <u>bit de signo</u> del resultado (fig. 5).

Tambièn hay que tener en cuenta que las operaciones que se hacen con ellos no afectan a los flags del registro de estado (màs adelante se explica la funciòn de estos flags)

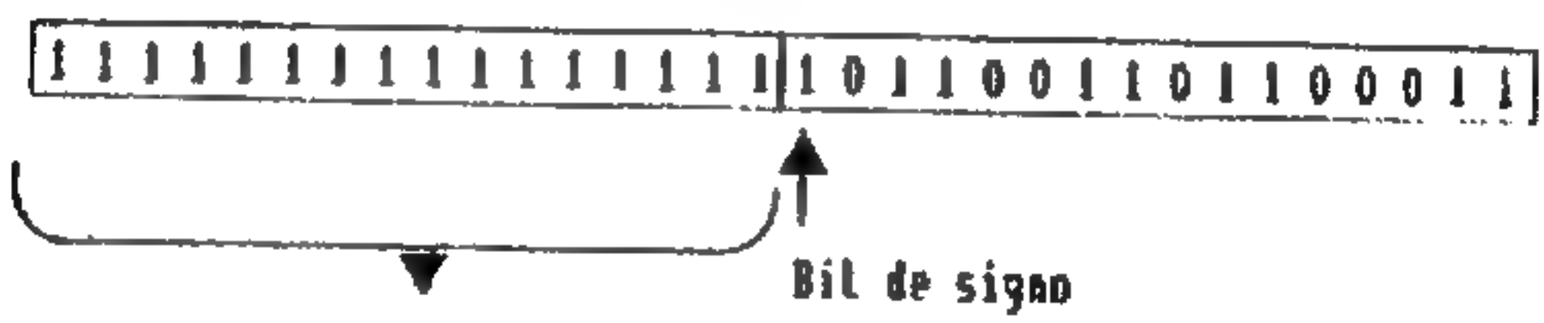


fig.5

9.- REGISTROS APUNTADORES DEL STACK

El stack es una zona de la memoria en la que se guarda información relativa al estado de la CPU.

Así pues los registros A7 y A7 son dos punteros que señalan el la dirección del último dato que contienen los stacks (usuario y supervisor).

El puntero A7 corresponde al stack usuario y el A7 al supervisor. En cada instante sólo uuciona uno de ellos, según el 68008 esté funcionando en uno u otro modo.

Generalmente cuando se funciona en modo usuario y hay un salto a una subrutina se usa el puntero del stack usuario. El stack supervisor se usa en modo supervisor al ocurrir unaarotura en la secuencia del programa (bien por una interrupción, o por un trap, como se verá más adelante).

La fig. 6 es una representación gráfica de un stack, comparable a una pila en la que se van introduciendo y sacando datos por la base, por lo que el primer dato en salir es siempre el último que se ha introducido.

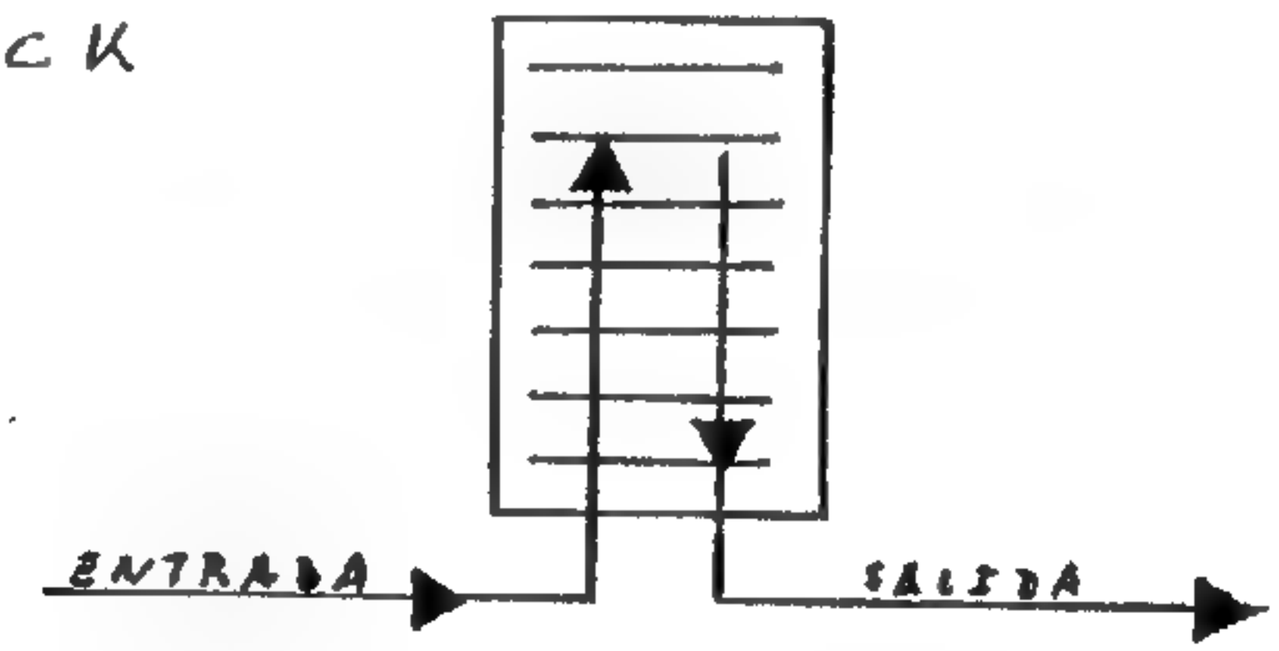


fig. 6

10.- CONTADOR DE PROGRAMA

Este registro (PC) se usa para contener la dirección de la siguiente palabra de instrucción a la que se va a acceder. Aunque la longitud del registro es de 32 bits, sólo se usan 20.

De estos 20 bits el bit n. 1 se emplea para determinar dentro de la dirección de la palabra accedida, si se refiere al byte par (LDS) o al impar (UDS).

Con 19 bits se puede direccionar 2^{19} = 524.288 palabras de 16 bits. Teniendo en cuenta la función del bit de menos peso, se puede direccionar sobre 1 Megabyte.

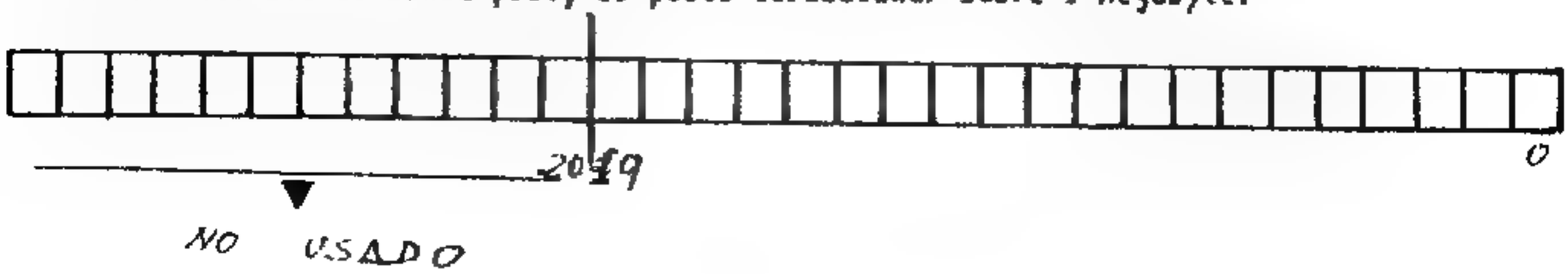


fig. 7

11.- REGISTRO DE ESTADO

Como su nombre indica, está destinado a guardar información del estado del microprocesador, y relativa a las operaciones que éste realiza en todo momento.

Este registro de 16 bits, se divide en dos bytes, uno destinado a guardar información referente al sistema, y otro al usuario.

Los 5 bits menos significativos del byte usuario son los "flags" o códigos de condición de la operación que ha realizado la última instrucción ejecutada por el 68008. Cada uno de estos bits ofrece información acerca de dicha operación. Los otros 3 bits del byte usuario no son significativos, y contienen siempre el valor 0.

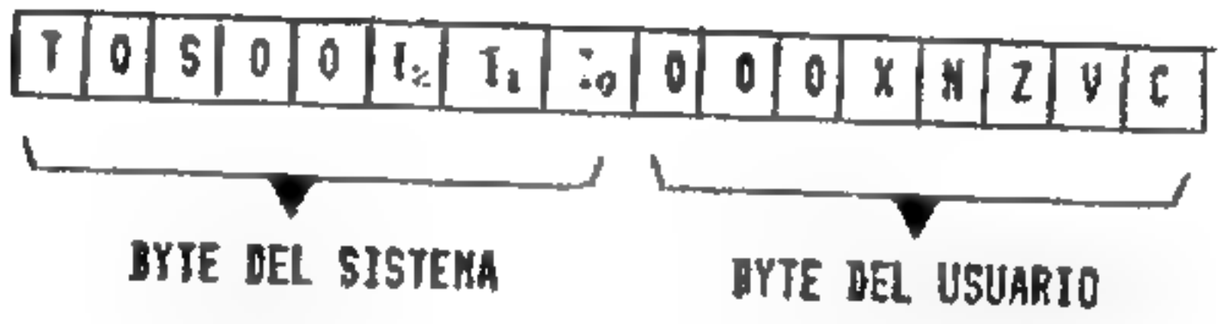


Fig. 8

La función de cada uno de estos bits es la siguiente:

C: ACARREO. Se pone a "1" si hay acarreo en los bits de más peso, al hacer una suma, o llevada en una resta. Veremos que también interviene en otras operaciones.
V: SOBREPASAMIENTO. Pasa a "1" cuando la magnitud de un resultado aritmético no puede ser representado en el registro por su tamaño. Sólo tiene sentido en operaciones en las que intervienen números con signo.
Z: CERO. Pasa a nivel "1" cuando el resultado de una operación ha sido cero.
N: SIGNO. Pasa a "1" si el resultado de una operación es negativo (sólo tiene sentido si se usan operandos con signo).
X: ACARREO EXTENDIDO. Presenta el mismo aspecto que "C" (acarreo). Se emplea en operaciones con múltiple precisión, para poder encadenar operaciones.

Los bites del byte del sistema sólo pueden alterarse en modo supervisor. Los tres bits menos significativos (I_0, I_2, I_3) componen los niveles de prioridad de la máscara de interrupciones (nivel de prioridad a partir del cual se atienden las interrupciones).

El bit "S" del byte del sistema, cambia entre el estado supervisor y el estado usuario. Cuando está a 1 el microprocesador trabaja en modo supervisor.

El bit "T" se usa para activar el "modo trazado" cuando está el bit a 1, lo cual sólo puede hacerse en modo supervisor.

Juan Palacio

NOVEDADES

ALSIPACK: La estrella de ALSI Comercial S.A.

En un reciente viaje que hice a Madrid realicè una visita a ALSI Comercial S.A.. Allì me mostraron, e hicieron una demostraciòn de su funcionamiento, lo que promete ser la gran estrella de la casa: ALSIPACK.

ALSIPACK es un paquete que revitaliza al Q.L. como un ordenador de gestiòn. En èl se aglutina todo lo mejor del software de gestiòn para el Q.L. realizado por ALSI, para nuestra sirva este botòn:

- Control de Almacenes
- Mailing
- Facturaciòn
- Albaranes
- Letras de cambio
- etc...
- Ficheros de Direcciones
- Control de pedidos
- Ofertas
- Contabilidad
- Copias

Este paquete se presenta en un sòlo disco de $3^1/_2$" y su precio serà de 96.000 pts.

Segùn me dijeron, en primicia para QLave, ALSIPACK estarà a la venta este mismo mes, ya que cuando vi el prototipo aùn estaban dàndole los ùltimos retoques.

Espero que este paquete de programas haga que el Q.L. se afiance màs en un sector del mercado que le corresponde por derecho propio.

==========

Tony Tebby, el diseñador del QDOS (sistema operativo para el Q.L.) y socio de honor de QLave, recientemente ha comenzado a comercializar productos para el Q.L. formando la Sociedad: QJump Ltd.

Lo ùltimo de QJump, cuya direcciòn es: 24 King Street Rampton, Cambridge CB4 4QD, England; es:

QL TOOLKIT II presentado en ROM al precio de 34.5 libras (V.A.T. incluido), junto con 2 libras para gastos postales.

QL MONITOR/DEBUGGER a la venta desde finales de Enero de 1986, su precio es de 19.95 libras (V.A.T. y gastos postales incluidos) en microdrive o en $5^1/_4$" floppy disc o 21.95 libras en $3^1/_2$" floppy disc.

EPROM PROGRAMMER a la venta desde finales de Enero de 1986, su precio es de 115 libras (V.A.T. y gastos postales incluidos).

QFLP UPGRADE ROM disponible al precio de 19.95 libras (V.A.T., gastos postales y manual incluidos). Aunque, por un periodo limitado de tiempo habrà un pequeño nùmero de unidades, sin empaquetar, con el manual en fotocopias por sòlo 14.95 libras.

==========

En pròximos nùmeros de QLave darè màs informaciòn sobre estos programas y os presentarè otras novedades para el Q.L.

S.O.Y.

TRUCOS Y RUTINAS...

En primer lugar quisiera corregir algún error que apareció en el programa que apareció en esta sección el mes pasado y que aunque éste corría perfectamente (razón por la que no reparé en ellos) se trata de algo a tener en cuenta.

En primer lugar la Macro que llamaba FINAL terminaba así:

```
ODOS    MT_FRJOB,1
MOVEQ   #0,D0
RTS
```

Pues bien ya que el programa era grabado con execute las dos últimas líneas son supérfluas, en estas situaciones el programa retorna al BASIC simplemente al cancelar el job. Ahora bien si el programa no hubiera sido relocalizable y se hubiera grabado con lbytes y llamado con un call si que hubiera sido necesario además de cancelar el job esas dos líneas que suprimimos.

En segundo lugar existió un error al pasar el programa en la revista la penúltima línea debe decir :

```
MENSAJE   DS.B    10
```

en lugar de DC. Ver que es un error de una letra pero que su significado es harto distinto DC significa una constante mientras que DS se refier a un espacio de trabajo que se reserva.

Vamos a comentar en este número algunos trucos que si bien pueden ser conocidos por muchos tal vez otros ignoren.

Así en la versión española se han incluido una serie de variables nuevas que se refieren al control de errores así como dos nuevos comandos.

Vosotros mismos, si tenéis ésta versión, podeis localizar estas variables en la lista que aparece en la zona de procedimientos residentes sin más que "peekear" en esta zona de la memoria. Decir que no suelen aparecer exactamente en el mismo lugar de la memoria por lo que tendréis que probar un poco si bien suelen quedar un poco por debajo de la dirección 260000 así pues para realizar esta exploración el programa a realizar sería similar a éste:

```
10 for f= 260000 to 250000 step -1
20 print f,chr$(peek(f))
30 next f
```

Las variables referentes a errores pueden tomar el valor 0 o 1 dependiendo de que se haya no se haya o si que se haya producido el error al que se refiere la variable.

Veamos a que se refieren cada una de estas variables:

ERR_NC	No completado.	ERR_NJ	Job inválido.
ERR_OM	No hay memoria.	ERR_OR	Fuera de rango.
ERR_BO	Memoria intermedia llena.	ERR_NO	No se encuentra el canal.

ERR_NF	No encontrado.	ERR_EX	Ya existe.
ERR_IU	En uso.	ERR_EF	Fin de fichero.
ERR_DF	Unidad llena.	ERR_BN	Nombre invàlido.

Estas nuevas variables se pueden usar conjuntamente con los nuevos comandos WHEN_____END WHEN y REPORT (CANAL) donde el cuerpo de la primera contendrà primero una expresiòn que contenga a una de las variables antes mencionadas y posteriormente un bloque de sentencias a ejecutar si se produce el còdigo de error indicado.

El segundo comando se refiere a sacar el mensaje de error por el canal especificado, asi sacaremos este por la ventana especifica que deseemos.

En el caso de que deseemos conocer cual es la direcciòn de memoria màs alta útil que poseemos nos basta con consultar a la variable del sistema SV_RAMT cuya direcciòn es 163872. Asi nos bastarà con teclear :

PRINT PEEK_L (163872)

Colocamos _L ya que se trata de una palabra larga.

Este valor variarà dependiendo sòlo de si poseemos la memoria base suministrada con el S.L. o si le hemos conectado una ampliaciòn de memoria.

Otra variable que puede tener su interès es SV_ARDEL que contiene el tiempo de auto-repeticiòn del teclado se trata de una palabra, por lo que lo que haremos es:

POKE_W 163980, valor

La variable SV_FSTAT se encarga de controlar el estado de parpadeo del cursor y que podremos alterar igualmente, el valor a introducir serà en este caso el siguiente:

POKE_W 164010, valor

Por otra parte si habèis explorado la lista anteriormente indicada de los comandos, funciones y variables del BASIC habrèis encontrado la funciòn VER$ de la que tampoco hace menciòn el manual en castellano y que en el caso de la versiòn española del ordenador deberà ser MGE , donde MG corresponde a la versiòn y E se refiere a España.

Quizà hayàis querido parar un listado sin tener que pulsar BREAK esto es posible realizarlo pulsando al mismo tiempo CTRL y F5. Detendrà cualquier proceso en curso hasta que pulsèis de nuevo cualquier tecla.

Por otra parte en el caso de que estemos trabajando con varios programas en multitarea que presenta sus resultados por diversas ventanas podremos cambiar el cursor activo de una ventana a otra sin màs que pulsar CTRL + C.

El editor incluido en la versiòn estàndar del S.L. es bastante incòmodo y sobre todo cuando se trata de realizar correcciones en diversas lìneas. Una posibilidad de desplazarnos màs deprisa por el listado es entrar en èl mediante EDIT y posteriormente a realizar la correcciòn en lugar de pulsar ENTER es posible usar los cursores para desplazarnos a través del listado hacia arriba o hacia abajo.

Isidro Asìn.

INFORMACION = LIBRERIA DE SOFWARE

La librería de programas está a disposición de todos los socios, así como esperamos que todos los socios contribuyan al crecimiento de la librería con sus programas.

Para realizar los envíos de vuestros programas a la librería debéis de cumplimentar y remitirnos junto con el programa un impreso como el que os adjuntamos. Esto lo hacemos así con vistas a tener derecho legal a distribuir esos programas entre los socios. Así pues se ceden los derechos del programa al club, pudiendo por supuesto su autor reclamarlos en cualquier momento (si por ejemplo desease venderlo....).

Para solicitar que se os envíe un programa de la librería deberéis mandar bien un cartucho, en el que os grabaremos el programa/s solicitado/s, más 70 ptas. en sellos de correos por gastos de envío certificado; o bien el dinero equivalente al cartucho, es decir 645 ptas. contrareembolso.

Las solicitudes irán dirigidas a: ANGEL ASIN BARCELONA (ver pag.2; librero).

Q L a v e
Club español independiente
de usuarios del QL.

EL SOCIO .....

PERTENECIENTE A QLave CERTIFICA QUE SIENDO POSEEDOR DE LOS DERECHOS DEL

PROGRAMA .....

LOS CEDE A QLave PUDIENDO RECUPERARLOS CUANDO LO SOLICITE.

Firma.

SUMARIO

1.- Portada

2.- Información sobre el club

3.- Editorial

4.- Correo de los socios

5.- Respuesta del Gobierno Civil

6.- Comentario de libros:
' Guía del usuario avanzado '

7.- Comentario de programas:
' Graphiql '
' Alsistock '

12.- Los 68000 : Una familia de microprocesadores

15.- Tonos y música en el B.L.

18.- Introducción al código máquina

28.- Novedades

29.- Trucos y rutinas

31.- Información sobre la librería de software

32.- Sumario